(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 February 2002 (07.02.2002)

PCT

(10) International Publication Number
WO 02/11389 A2

(51) International Patent Classification[7]: H04L 29/00

(21) International Application Number: PCT/US01/22312

(22) International Filing Date: 16 July 2001 (16.07.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/627,723 28 July 2000 (28.07.2000) US

(71) Applicant: DIALPAD COMMUNICATIONS, INC. [US/US]; 2953 Bunker Hill Lane, Suite 400, Santa Clara, CA 95054 (US).

(72) Inventors: CHO, Wongyu; 7013 Silver Star Court, San Jose, CA 95120 (US). HONG, Hyungkeun; 2370 Homestead Road, Apt. 10, Santa Clara, CA 95050 (US).

(74) Agents: COOK, Carmen, C. et al.; Skjerven Morrill MacPherson LLP, 25 Metro Drive, Suite 700, San Jose, CA 95110 (US).

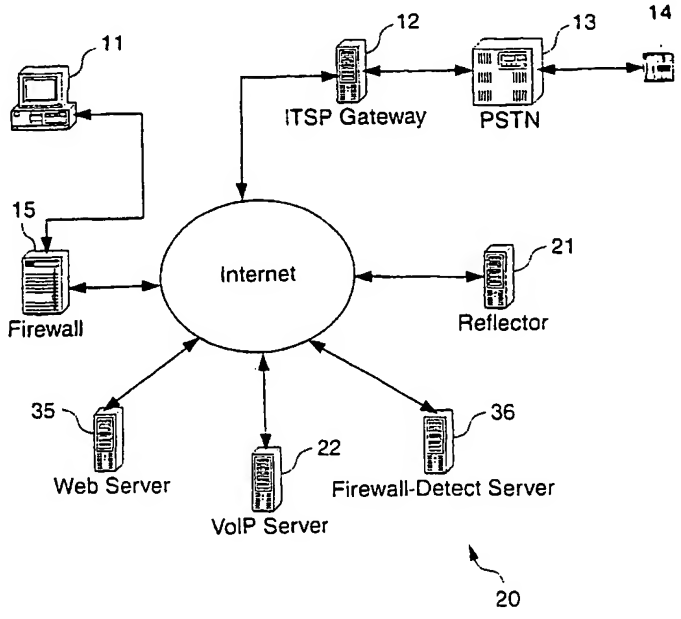
(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: DATA EXCHANGE WITH COMPUTERS WITHIN A SECURE NETWORK



(57) Abstract: A first computer transmits data to a second computer in accordance with a first protocol. The second computer receives the data and transmits them to a third computer, which is within a secure network, using a second protocol that is different from the first protocol. The data is transmitted to the third computer over a connection originated from within the secure network, thereby allowing the data to pass through network security devices such as firewalls.

WO 02/11389 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

DATA EXCHANGE WITH COMPUTERS WITHIN A SECURE NETWORK

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention generally relates to computer networks and more particularly to methods and associated apparatus for exchanging data with computers within a secure network.

2. Description of the Related Art

Voice Over Internet Protocol (VoIP) is a technique for transmitting voice signals using the Internet Protocol (IP). In VoIP, voice signals from an audio communications device (e.g., a regular telephone or a computer equipped with audio peripherals) are encapsulated in packets and transmitted as voice data over a network such as the Internet.

FIG. 1A shows a schematic diagram of an exemplary VoIP network. A local user using a computer 11 equipped with a sound card and headset, for example, provides voice data to an Internet Telephone Service Provider (ITSP) gateway 12 over the Internet. ITSP gateways are available from several network service providers including the IDT Corporation and Qwest Communications. ITSP gateway 12 is coupled to a remote user who, in this example, uses a regular telephone 14 linked to a public switched telephone network (PSTN) 13. PSTN 13 provides either wireline or wireless telephone service commonly known as "plain old telephone service" (POTS). ITSP gateway 12 converts the voice data from computer 11 into corresponding voice signals for transmission to telephone 14 through PSTN 13. Conversely, ITSP gateway 12 converts voice signals

received from telephone 14 into a form that is suitable for transmission over the Internet to computer 11.

FIG. 1B schematically illustrates a network wherein computer 11 is located behind a firewall 15. Firewalls are well known network components for screening incoming data to a secure network. Data that use a connection created by computers behind firewall 15 are able to pass through firewall 15. A connection is a communications link between two application programs (e.g., programs running on separate computers) on a network. The connection is identified by the IP addresses and port numbers of the two connected application programs. The IP address identifies the computer on which an application program runs while the port number identifies the application program in the computer. If computer 11 creates a connection through firewall 15 to ITSP gateway 12, ITSP gateway 12 can send data to computer 11 using the same connection (unless of course firewall 15 is specifically configured to block any data from ITSP gateway 12). However, computers outside firewall 15 cannot arbitrarily create a connection through firewall 15. Thus, unless ITSP gateway 12 uses a connection originated from behind firewall 15, voice signals from telephone 14 will not reach computer 11.

Most ITSP gateways send and receive voice data in accordance with the Real-Time Transport Protocol (RTP), which utilizes User Datagram Protocol (UDP) packets. Both RTP and UDP are well known. RTP is described in the Internet Engineering Task Force (IETF) documents RFC 1889, "RTP: A Transport Protocol For Real-Time Applications" and RFC 1890, "RTP Profile For Audio And Video Conferences With Minimal Control". UDP and TCP/IP are described in "TCP/IP Illustrated Volume 1", W. R. Stevens, Addison Wesley 1994. The aforementioned references are incorporated herein by reference in their entirety.

The use of UDP to transmit data to a computer within a secure network presents a problem because UDP does not use a pre-established connection in transmitting data. Thus, in most situations, the firewall will not allow incoming UDP packets from entering the secure network.

SUMMARY

The present invention relates to a method and associated apparatus for exchanging data with computers within a secure network.

In one embodiment, a first computer transmits data to a second computer in accordance with a first protocol. The second computer receives the data and transmits them to a third computer, which is within a secure network, using a second protocol that is different from the first protocol. The data is transmitted to the third computer over a connection originated from within the secure network, thereby allowing the data to pass through network security devices such as firewalls. In one specific example, the first protocol is the User Datagram Protocol (UDP) and the second protocol is the Transport Control Protocol (TCP).

These and other features of the invention will be apparent to persons of ordinary skill in the art upon reading the following description and figures.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A shows a schematic diagram of a network in the prior art.

FIG. 1B shows a schematic diagram of a network in the prior art which includes a firewall.

FIG. 2 shows a schematic diagram of a network in accordance with an embodiment of the invention.

FIG. 3 shows a method for transmitting data to a computer within a secure network in one embodiment.

FIG. 4 shows a method for transmitting data to a computer within a secure network in another embodiment.

FIGS. 5A, 5B, and 5C show schematic representations of TCP packets in one embodiment.

FIG. 6 shows a flow diagram of a VoIP telephone call in one embodiment.

FIG. 7 shows a schematic representation of software processes for transmitting data to a computer within a secure network in one embodiment.

The use of the same reference symbol in different figures indicates the same or identical elements.

DETAILED DESCRIPTION

FIG. 2 shows a schematic diagram of a network 20 wherein a reflector 21 application program, running on a computer acting as a server, is utilized to allow data transmission through firewall 15. While reflector 21 can be resident on any computer in network 20 that is outside firewall 15, reflector 21 is preferably on a separate high performance computer located close to ITSP gateway 12 to minimize data transmission delay and possible data loss. Network 20 includes a VoIP server 22 for setting up a VoIP telephone call between the user on computer 11 and the user on telephone 14. Reflector 21 can be employed independent of VoIP server 22, and can be generally used to exchange data with computers behind firewalls.

Referring to FIG. 2, client program for making the VoIP telephone call and files containing information about network 20 can be downloaded from a web server 35 (e.g., a website). Web server 35 can be a conventional file server or any of the VoIP portals accessible over the Internet such as those from Dialpad.com, Inc. of Santa Clara, California. Network 20 also includes a firewall-detect server 36 which, as discussed further below, enables a client program running on computer 11 to detect whether it

is behind a firewall. It is to be noted that client-server architectures, in general, are well known.

VoIP server 22, ITSP gateway 12, web server 35, and the client program running on computer 11 can also be of the same type as the scaleable communications system disclosed in U.S. Patent Application No. 09/401,898, "Scaleable Communications System," filed on September 24, 1999, incorporated herein by reference in its entirety. Reflector 21 can also be used with VoIP systems and services accessible over the Internet such as those from Dialpad.Com, Inc.

FIG. 3 shows a method for transmitting data to a computer within a secure network in one embodiment. In action 30, multimedia data (e.g., voice, video, still images, and/or fax) from a source server such as ITSP gateway 12 are transmitted to reflector 21 in accordance with a first protocol. Reflector 21 receives the first protocol packets from the source server (action 31), extracts the multimedia data from the first protocol packets, and encapsulates the multimedia data in accordance with a second protocol (action 32). In one example, the multimedia data are formatted in accordance with RTP and the first protocol is UDP. Reflector 21 transmits the second protocol packets to an application (e.g., client program) behind the firewall (action 33), where the multimedia data are extracted (action 34). In one example, the second protocol is the Transport Control Protocol (TCP). TCP, a connection-oriented protocol, transports data using a pre-established connection between two application programs. Thus, reflector 21 can transmit data to an application program behind the firewall by utilizing a TCP connection originated from within the secure network. TCP, in general, is well known; e.g., see the incorporated reference "TCP/IP Illustrated Volume 1", W. R. Stevens, Addison Wesley 1994. TCP/IP software, commonly known as TCP/IP protocol stack, is also

commercially available from several vendors including Sun Microsystems.

FIG. 4 shows a method for transmitting voice data from ITSP gateway 12 to computer 11 of network 20 (FIG. 2) in one embodiment. In action 39, a client program running on computer 11 transmits a UDP packet to firewall-detect server 36 located outside firewall 15. Firewall-detect server 36 is a server that waits for a UDP packet from the client program and correspondingly replies with another UDP packet. The UDP packets transmitted to and from firewall-detect server 36 are intended to determine whether the client program runs on a computer behind a firewall. In action 40, the client program waits for a reply from firewall-detect server 36. If the client program receives a reply from firewall-detect server 36, the client program is not behind a firewall and reflector 21 is therefore not required (action 41). In this example, however, the client program is behind firewall 15, which blocks the reply from firewall-detect server 36. After failing to receive a reply from firewall-detect server 36 within a predetermined amount of time, the client program concludes that it must be behind a firewall and accordingly creates a conventional TCP connection to reflector 21 (action 42). Any protocol suitable for transmission through a firewall, such as those that utilize a pre-established connection, can be used instead of TCP. The IP address of reflector 21 can be hard-coded in the client program or downloaded from web server 35.

In action 43, reflector 21 provides an RTP port number to the client program. This RTP port number along with reflector 21's IP address, as explained below, will eventually be provided to ITSP gateway 12 so that RTP data can be transmitted from ITSP gateway 12 to reflector 21. Hereinafter, the IP address and RTP port number of reflector 21 for receiving RTP data from ITSP gateway 12 are collectively referred to as reflector 21's RTP transport address. In action 44, the client program

provides reflector 21's RTP transport address to VoIP server 22, and obtains from VoIP server 22 the transport address of ITSP gateway 12. The transport address of ITSP gateway 12 consists of the IP address and RTP port number that ITSP gateway uses to receive RTP data from reflector 21.

In action 45, the client program provides reflector 21 the RTP transport address of ITSP gateway 12. This allows reflector 21 to transmit the RTP data it receives from the client program to ITSP gateway 12.

In action 46, VoIP server 22 provides the RTP transport address of reflector 21 to ITSP gateway 12. Action 46 typically occurs during the time the VoIP telephone call between computer 11 and telephone 14 is being setup by VoIP server 22 and ITSP gateway 12 in accordance with the International Telecommunication Union (ITU) H.323 standard and associated protocols. ITU H.323 is well known; e.g., see ITU-T Recommendation Q.931, ITU-T Recommendation H.245, and ITU-T Recommendation H.323, all incorporated herein by reference.

In action 47, ITSP gateway 12 creates RTP data channels to and from reflector 21 in accordance with ITU H.323. Note that both ITSP gateway 12 and reflector 21 know each other's RTP transport address and can thus exchange RTP data over the RTP data channels. ITSP gateway 12 formats the voice signals from telephone 14 in accordance with RTP (hereinafter "RTP data"), encapsulates the RTP data in UDP packets, and transmits the UDP packets over the RTP data channel from ITSP gateway 12 to reflector 21 (action 48). The flow of RTP data between ITSP gateway 12 and reflector 21 over the RTP data channels is also known as an RTP data stream.

In action 49, reflector 21 extracts the RTP data from the UDP packets received from ITSP gateway 12. The RTP data are then encapsulated in TCP packets before being

transmitted to the client program on computer 11. FIG. 5A shows a schematic representation of a TCP packet 61 suitable for transmission from reflector 21 to the client program in one example. TCP packet 61 is a conventional TCP packet containing the ASCII characters "R", "T", "P", "D", "A", "T", and "A" in the first 7 bytes of its data section. A byte 62 indicates the size of the following RTP data. An ASCII character "\r" (backslash-r) follows byte 62. The first 10 bytes of the data section of TCP packet 61, also referred to as a boundary portion, informs the client program that TCP packet 61 contains RTP data and that the bytes following ASCII character "\r" are RTP data having a size indicated in byte 62.

In action 50, reflector 21 transmits the TCP packets containing RTP data to the client program in computer 11 over the TCP connection previously established in action 42. Because that TCP connection was created by the client program, which is in the secure network, reflector 21 is able to transmit the TCP packets through firewall 15.

In action 51, the client program extracts the RTP data from the TCP packets. Thereafter, the client program processes the RTP data by playing the corresponding voice information from telephone 14 (action 52).

The transmission of RTP data from computer 11 to telephone 14 is performed using a process similar to that shown in FIG. 4 except in the opposite direction. In one example, the client program formats the voice of the local user in accordance with RTP and encapsulates the resulting RTP data in TCP packets, which are then transmitted to reflector 21 using the TCP connection previously established in action 42. FIGS. 5B and 5C show a schematic representation of TCP packets 63 and 65 suitable for transmission from the client program to reflector 21 in one example. TCP packet 63 is a conventional TCP packet with the ASCII characters "R", "T", "P", "D", "A", "T", "A", and "SPACE" (i.e., a space bar character) in the

first 8 bytes of its data section. TCP packet 63 informs reflector 21 that another TCP packet, TCP packet 65, containing the actual RTP data will be transmitted from the client program. A byte 64 of TCP packet 63 indicates the size of the RTP data in TCP packet 65. In TCP packet 63, there are no relevant data following the "\r" character.

Referring to FIG. 5C, TCP packet 65 is transmitted from the client program in computer 11 to reflector 21 following the transmission of TCP packet 63. Reflector 21 extracts the RTP data from received TCP packets 65 and encapsulates the RTP data in UDP packets for transmission over the RTP data channel from reflector 21 to ITSP gateway 12. ITSP gateway 12 then extracts the RTP data from the UDP packets and relays the voice information to telephone 14.

Voice data from computer 11 can also be directly transmitted to ITSP gateway 12 because computer 11 is behind firewall 15, and thus can create another connection through firewall 15 onto ITSP gateway 12. In one example, the client program formats the user's voice in accordance with RTP and encapsulates the resulting RTP data in UDP packets. The client program directly transmits the UDP packets to ITSP gateway 12 without going through reflector 21. ITSP gateway 12 extracts the RTP data from the UDP packets and relays the voice information to telephone 14.

FIG. 6 shows a flow diagram of an exemplary VoIP telephone call between computer 11 and telephone 14 in network 20 (FIG. 2). In flow 69, the client program on computer 11 transmits a UDP packet to firewall-detect server 36 to determine if computer 11 is behind a firewall. All communications between the client program and firewall-detect server 36 are over an arbitrary UDP connection. Because computer 11 is behind firewall 15 in this example, the client program will not receive a response from firewall-detect server 36. In flow 70, the

client program thus makes a TCP connection, hereinafter referred to as TCP connection "A", to reflector 21. All communications between the client program and reflector 21 are over TCP connection "A". Also during flow 70, reflector 21 provides its RTP transport address to the client program.

In flow 71, the client program makes a separate TCP connection, hereinafter referred to as TCP connection "B", to VoIP server 22 and informs VoIP server 22 that it wants to make a VoIP telephone call to telephone 14. All communications between the client program and VoIP server 22 are over TCP connection "B". In flow 71, the client program also provides reflector 21's RTP transport address to VoIP server 22. In flow 72, VoIP server 22 informs the client program that the VoIP telephone call is proceeding.

In flow 73, VoIP server 22 setups the VoIP telephone call with ITSP gateway 12 in accordance with the ITU H.323 standard. All communications between VoIP server 22 and ITSP gateway 12 are over a separate TCP connection hereinafter referred to as TCP connection "C". In flow 74, ITSP gateway 12 makes a call to telephone 14 via PSTN 13 (FIG. 2) and receives a ring signal. In flow 75, ITSP gateway 12 informs VoIP server 22 that telephone 14 has been contacted. VoIP server 22 receives the RTP transport address of ITSP gateway 12 at this time. In flow 76, VoIP server 22 relays the information to the client program, which now knows that the telephone 14 is ringing and can be picked-up by the remote user at any time. Also in flow 76, the client program receives the RTP transport address of ITSP gateway 12 from VoIP server 22.

In flow 77, ITSP gateway 12 informs VoIP server 22 that telephone 14 has been picked-up by the remote user and that it will start transmitting RTP data to reflector 21 (using reflector 21's RTP transport address) over an RTP data channel. There are two RTP data channels in this example, which are an RTP data channel from ITSP gateway

12 to reflector 21 and another RTP data channel from reflector 21 to ITSP gateway 12. In flow 78, VoIP server 22 informs the client program that telephone 14 has been picked up and that the client program can now send and receive RTP data via reflector 21.

In flow 79, the client program reports its status (including error conditions and whether it is still making the VoIP telephone call) to VoIP server 22. Flow 79 is periodically performed while the VoIP telephone call is in progress. In one example, VoIP server 22 will terminate an in progress VoIP telephone call if VoIP server 22 ceases to receive a status from the client program.

In flow 80, the client program informs reflector 21 that the remote user has picked-up telephone 14 and that reflector 21 should expect to receive RTP data over the RTP data channel from ITSP gateway 12. Also in flow 80, reflector 21 receives the RTP transport address of ITSP gateway 12 from the client program. This allows reflector 21 to send RTP data over the RTP data channel to ITSP gateway 12.

In flow 81, RTP data representing voice information are transported between reflector 21 and ITSP gateway 12 using UDP packets over the RTP data channels. In flow 82, the RTP data are transported between reflector 21 and the client program using TCP packets over TCP connection "A".

In flow 83, DTMF touch tone signals, if any, are transmitted from the client program to VoIP server 22. The DTMF touch tone signals are then relayed by VoIP server 22 to ITSP gateway 12 over TCP connection "C" (not shown).

In flow 84, the client program informs reflector 21 that the user on computer 11 decides to terminate the VoIP telephone call. In flow 85, the client program also informs VoIP server 22 that the VoIP telephone call is

being terminated. In flow 86, VoIP server 22 accordingly informs ITSP gateway 12 to close the RTP data channels between ITSP gateway 12 and reflector 21. In flow 87, ITSP gateway 12 informs VoIP server 22 that the RTP data channels have been closed. In flow 88, VoIP server 22 informs the client program that the VoIP telephone call has been terminated.

One of ordinary skill in the art will appreciate that the sequence of events in the flow diagram of FIG. 6 can be re-arranged without detracting from the merits of the invention. For example, the RTP transport address of ITSP gateway 12 can be provided to reflector 21 at any time before flow 82, and not necessarily during flow 80 when the client program provides its status to reflector 21.

In one embodiment, reflector 21 is written in the "C" programming language and runs on a SPARC™ Station computer with the Solaris™ operating system, both of which are available from Sun Microsystems. Of course, other programming languages, computers, and operating systems can also be used.

FIG. 7 shows a schematic representation of the relevant reflector 21 processes and their relationship with ITSP gateway 12 and the client program on computer 11. Tables 1 and 2, which show pseudo-codes FWMAIN.C and FWGATE.C, are discussed with reference to FIG. 7.

TABLE 1. FWMAIN.C (PARENT PROCESS)

```
Main(){
 Call RunFWGServer();
}
RunFWGServer(){
     Create socket pairs for communication between child
          and parent processes;
     Fork as many child processes as specified;
     While(){
     Wait for connection from Client;
```

```
            If a connection is established, assign the
                processing of the connection to one of the
                Child processes;
        }
    }
```

TABLE 2. FWGATE.C (CHILD PROCESSES, THREAD PAIRS)

```
ChildProcess(){
    While(){
        Wait for notification from Parent process that a
            Client connection has been established;
        If the notification arrives, create two Threads
            for serving the corresponding Client;
    }
}
ThreadProcR(){//Thread-1
    Allocate an RTP port number;
    Send the RTP port number to Client;
    Wait for Client to provide the IP address and RTP
        port number of ITSP gateway; //(This is not
        needed if RTP data are to be transmitted
        directly from Client to ITSP gateway)
    Wait for the establishment of an RTP data channel
        between the reflector and ITSP gateway;
    Wait for notification that the Client is ready to
        accept RTP data;
    //(The following While loop is not needed if RTP data
        are to be transmitted directly from Client to
        ITSP gateway)
    While () {
        Receive TCP packets from Client;
        Extract the RTP data from the TCP packets
            received from Client;
        Encapsulate the RTP data in UDP packets;
        Transmit the UDP packets to ITSP gateway;
    }
}
ThreadProcToCli(){//Thread-2
    While(){
```

```
                Receive UDP packets from the ITSP gateway;
                Extract the RTP data from the UDP packets
                    received from the ITSP gateway;
                Encapsulate the RTP data in TCP packets;
                Send the TCP packets to the Client;
            }
        }
```

FWMAIN.C includes the pseudo-code for the parent process illustrated in FIG. 7. As shown in Table 1, the parent process creates multiple child processes, with each child process communicating with the parent process using a socket pair provided by the operating system via a system call. A socket pair is a communications link between two processes running on the same computer. In one example, the parent process creates ten child processes. The parent process then waits for the client program to establish a TCP connection to reflector 21. Once a TCP connection is established, the parent process assigns the processing of the connection to one of the child processes.

FWGATE.C includes the pseudo code for the child processes, thread-1, and thread-2 illustrated in FIG. 7. As shown in Table 2, each child process creates a thread pair, which are thread-1 and thread-2, upon notification from the parent process that a TCP connection from the client program has been established. A thread pair is created for each TCP connection between a client program and reflector 21. In one example, each child process supports up to 500 thread pairs. By allocating tasks to child processes and threads, reflector 21 can efficiently process multiple TCP connections.

Referring to the procedure ThreadProcR() shown in Table 2, thread-1 allocates an RTP port number and sends it to the client program. Thread-1 then waits for the client program to provide the IP address and RTP port number of ITSP gateway 12, for the establishment of an RTP

data channel between reflector 21 and ITSP gateway 12, and for notification that the client program is ready to accept RTP data. In the case where RTP data are not directly transmitted from the client program to ITSP gateway 12, thread-1 receives TCP packets from the client program, extracts the RTP data from received TCP packets, encapsulates the RTP data in UDP packets, and transmits the UDP packets to ITSP gateway 12.

Still referring to Table 2, thread-2 (procedure ThreadProcToCli()) receives UDP packets from ITSP gateway 12, extracts the RTP data from the received UDP packets, encapsulates the RTP data in TCP packets, and transmits the TCP packets to the client program. Note that TCP/IP operations are generally transparent to the applications programmer because the processing of packets (including data extraction, encapsulation, transmission, and reception) are performed by the TCP/IP protocol stack via library calls.

A method and apparatus for exchanging data with computers in a secure network have been disclosed. While specific embodiments of this invention have been described, it is to be understood that these embodiments are illustrative and not limiting. Many additional embodiments that are within the broad principles of this invention will be apparent to persons skilled in the art. For example, the invention can be used to transmit any type of data, not just voice data, to computers within a secure network. Further, the invention is applicable to any type of network, including those not linked to the Internet.

CLAIMS

What is claimed is:

1. A method for transmitting data comprising the acts of:

transmitting data from a first computer outside a secure network to a second computer outside said secure network in accordance with a first protocol; and

transmitting said data from said second computer to a third computer inside said secure network in accordance with a second protocol that is different from said first protocol.

2. The method of claim 1 wherein said data include information selected from a group consisting of voice, video, still image, and fax.

3. The method of claim 1 wherein said data are in a format conforming to the Real-Time Transport Protocol (RTP).

4. The method of claim 3 wherein said data include voice information.

5. The method of claim 4 wherein a firewall separates said third computer from said first and second computers.

6. The method of claim 5 wherein said first protocol is the User Datagram Protocol (UDP) and said second protocol is the Transport Control Protocol (TCP).

7. The method of claim 1 wherein said first protocol is the User Datagram Protocol (UDP) and said second protocol is the Transport Control Protocol (TCP).

8. The method of claim 7 wherein said data conform to the Real-Time Transport Protocol (RTP).

9. A system for transmitting data to computers inside a secure network comprising:

a first computer outside said secure network;

a second computer outside said secure network and communicating with said first computer in accordance with a first protocol; and

a third computer inside said secure network and communicating with said second computer in accordance with a second protocol that is different from said first protocol, whereby data is transmitted from said first computer to said third computer through said second computer.

10. The system of claim 9 further comprising a firewall coupled between said third computer and said second computer.

11. The system of claim 10 wherein said first protocol is the User Datagram Protocol (UDP) and said second protocol is the Transport Control Protocol (TCP).

12. The system of claim 10 wherein said data conform to the Real-Time Transport Protocol (RTP).

13. The system of claim 9 wherein said first computer is an Internet Telephone Service Provider gateway.

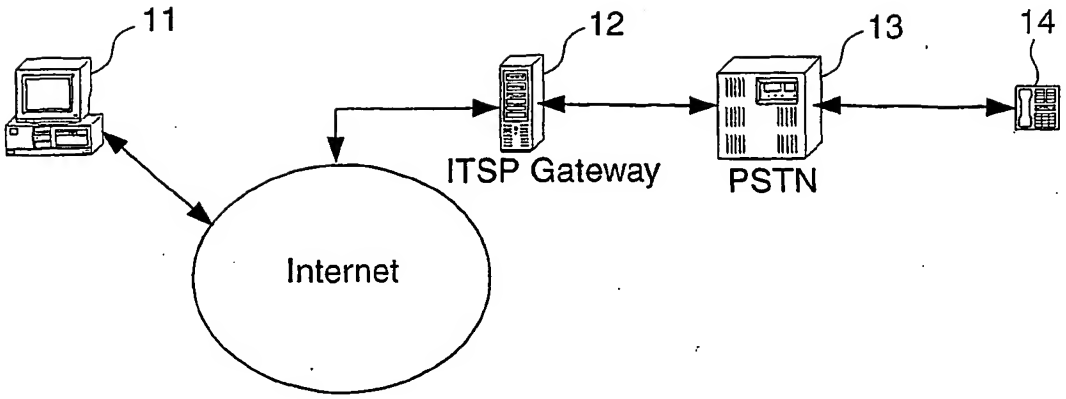


FIG. 1A
(Prior Art)

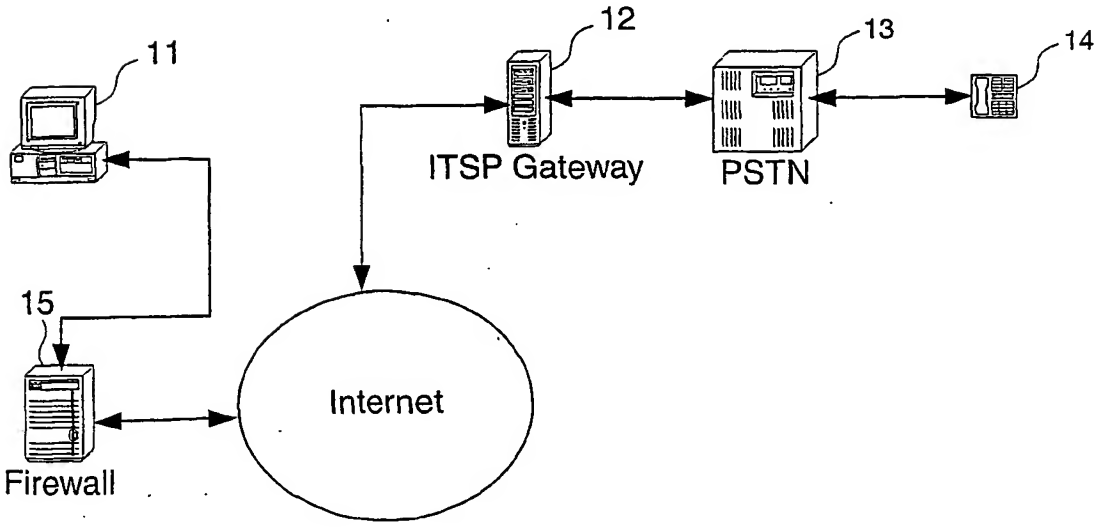


FIG. 1B
(Prior Art)

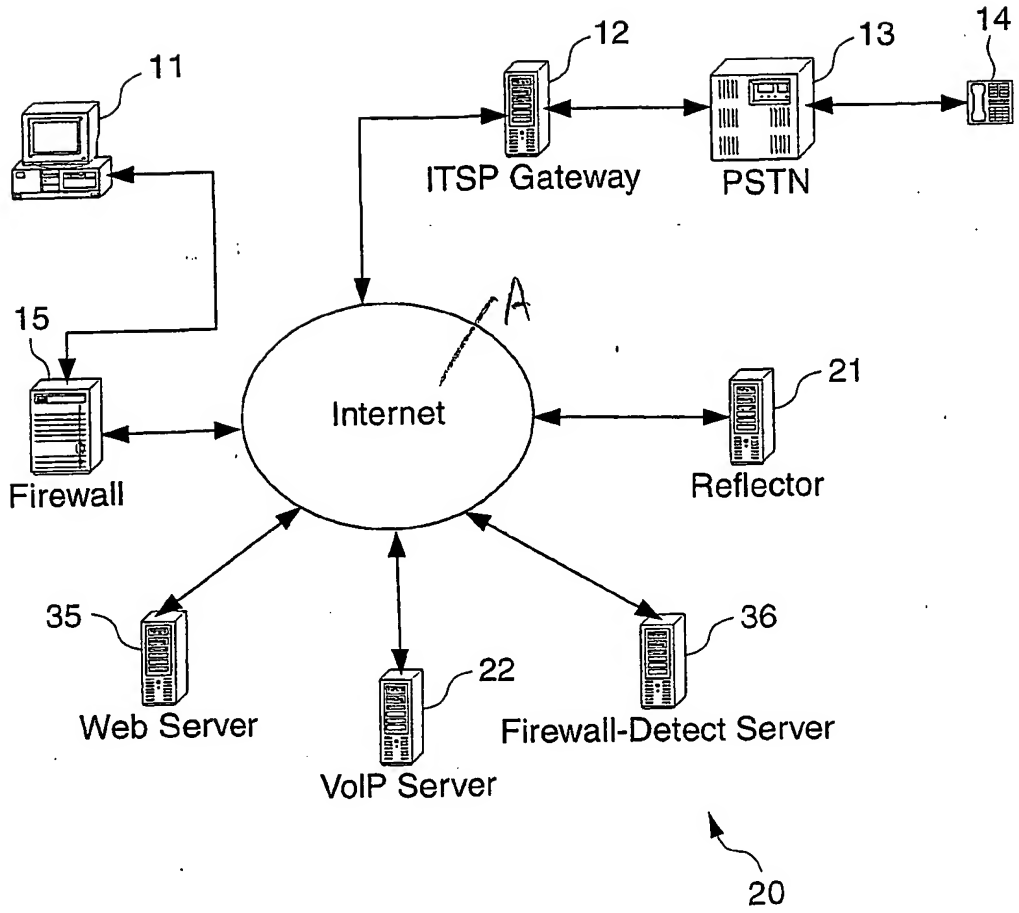

11
12
13
14
ITSP Gateway
PSTN
15
A
Internet
21
Firewall
Reflector
35
36
22
Web Server
Firewall-Detect Server
VoIP Server
20

FIG. 2

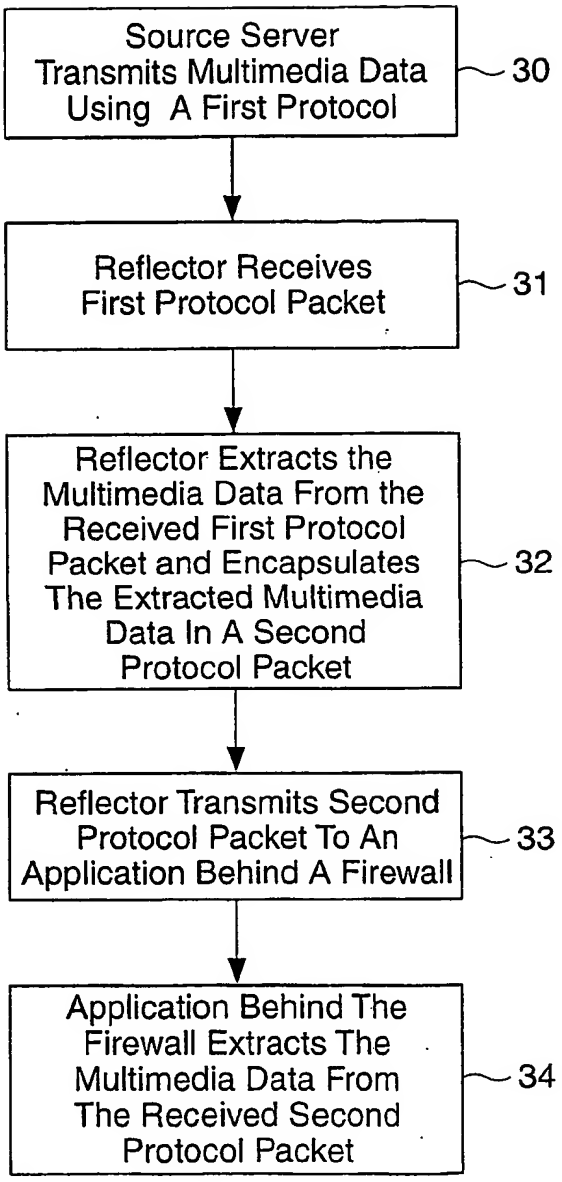

Source Server Transmits Multimedia Data Using A First Protocol
30
Reflector Receives First Protocol Packet
31
Reflector Extracts the Multimedia Data From the Received First Protocol Packet and Encapsulates The Extracted Multimedia Data In A Second Protocol Packet
32
Reflector Transmits Second Protocol Packet To An Application Behind A Firewall
33
Application Behind The Firewall Extracts The Multimedia Data From The Received Second Protocol Packet
34

FIG. 3

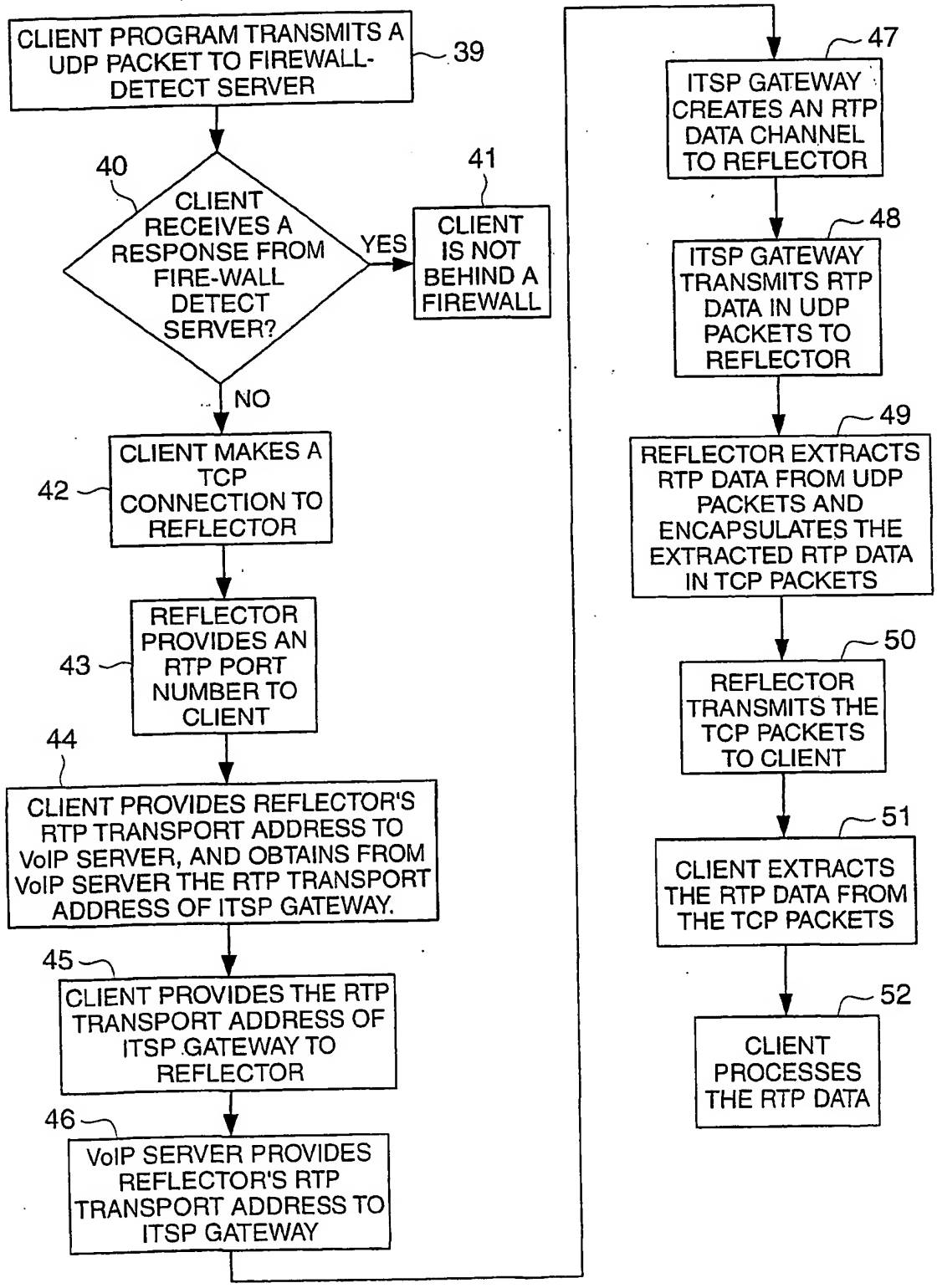

CLIENT PROGRAM TRANSMITS A UDP PACKET TO FIREWALL-DETECT SERVER
39
40
CLIENT RECEIVES A RESPONSE FROM FIRE-WALL DETECT SERVER?
YES
41
CLIENT IS NOT BEHIND A FIREWALL
NO
42
CLIENT MAKES A TCP CONNECTION TO REFLECTOR
43
REFLECTOR PROVIDES AN RTP PORT NUMBER TO CLIENT
44
CLIENT PROVIDES REFLECTOR'S RTP TRANSPORT ADDRESS TO VoIP SERVER, AND OBTAINS FROM VoIP SERVER THE RTP TRANSPORT ADDRESS OF ITSP GATEWAY.
45
CLIENT PROVIDES THE RTP TRANSPORT ADDRESS OF ITSP GATEWAY TO REFLECTOR
46
VoIP SERVER PROVIDES REFLECTOR'S RTP TRANSPORT ADDRESS TO ITSP GATEWAY
47
ITSP GATEWAY CREATES AN RTP DATA CHANNEL TO REFLECTOR
48
ITSP GATEWAY TRANSMITS RTP DATA IN UDP PACKETS TO REFLECTOR
49
REFLECTOR EXTRACTS RTP DATA FROM UDP PACKETS AND ENCAPSULATES THE EXTRACTED RTP DATA IN TCP PACKETS
50
REFLECTOR TRANSMITS THE TCP PACKETS TO CLIENT
51
CLIENT EXTRACTS THE RTP DATA FROM THE TCP PACKETS
52
CLIENT PROCESSES THE RTP DATA

FIG. 4

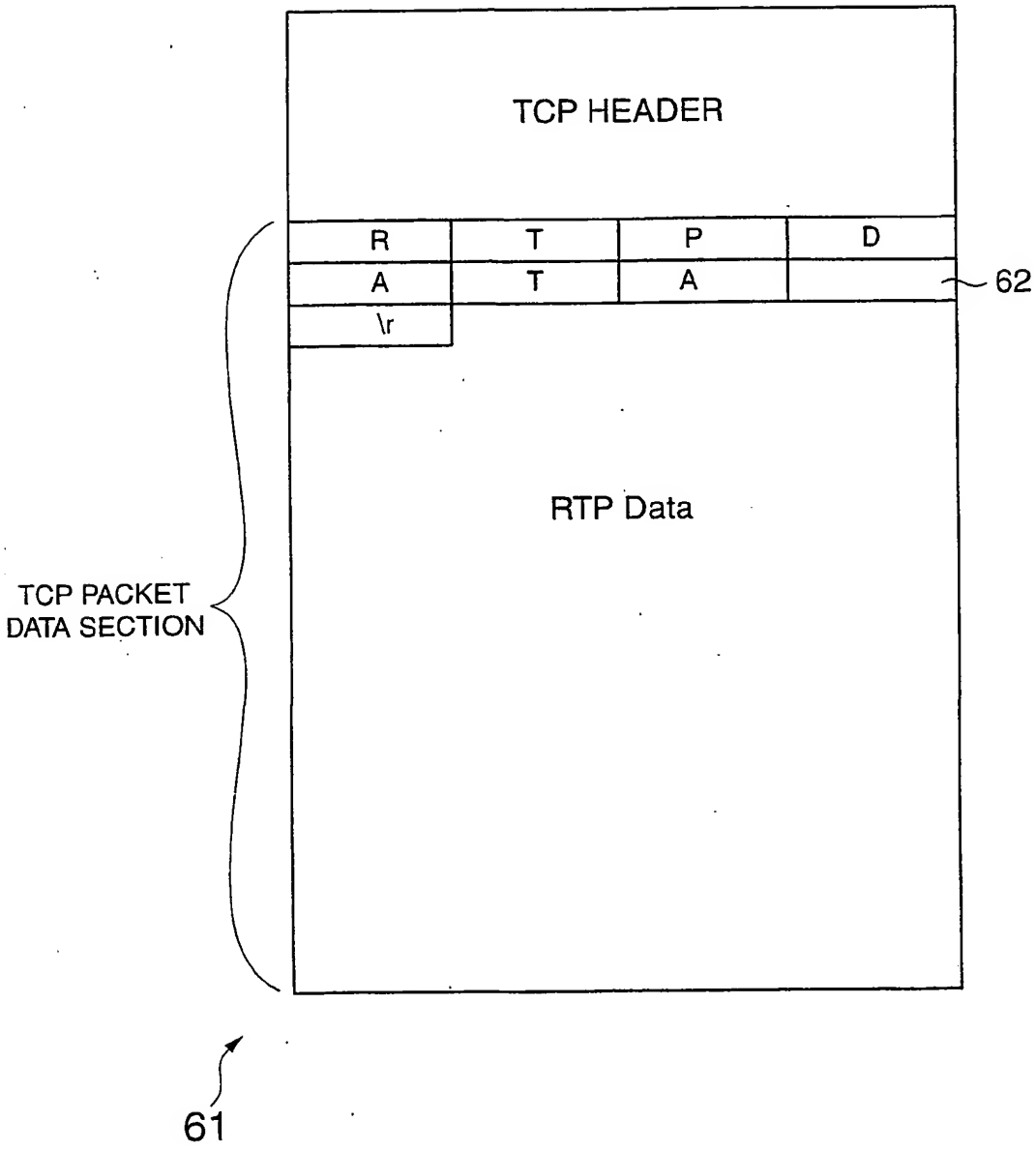


FIG. 5A

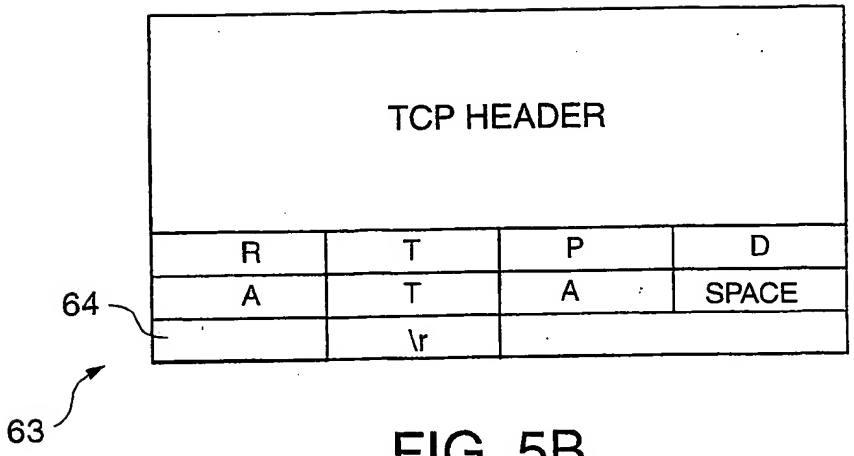


FIG. 5B

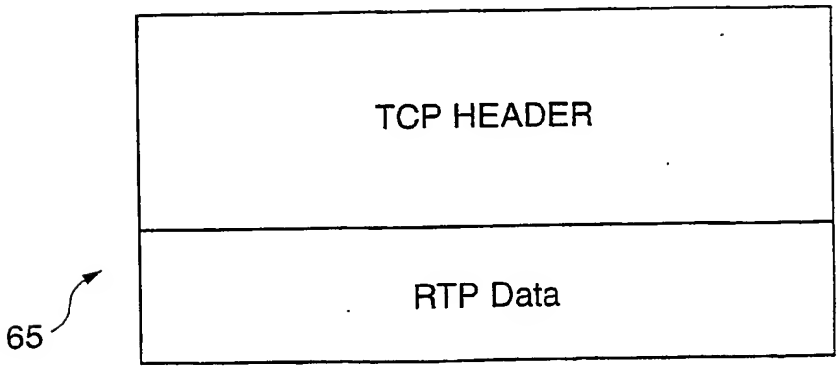


FIG. 5C

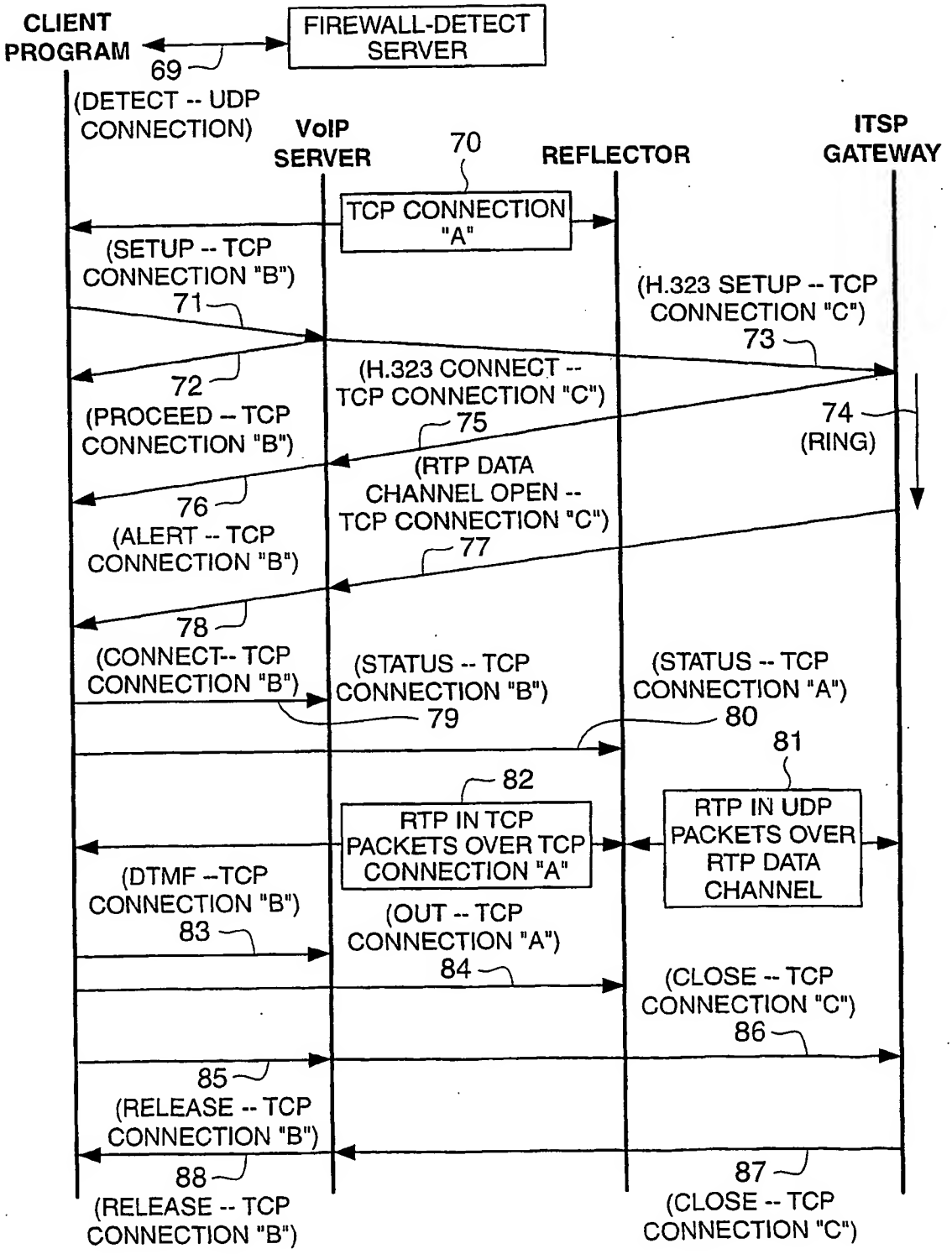

CLIENT PROGRAM
FIREWALL-DETECT SERVER
69
(DETECT -- UDP CONNECTION)
VoIP SERVER
70
REFLECTOR
ITSP GATEWAY
TCP CONNECTION "A"
(SETUP -- TCP CONNECTION "B")
71
(H.323 SETUP -- TCP CONNECTION "C")
73
72
(PROCEED -- TCP CONNECTION "B")
(H.323 CONNECT -- TCP CONNECTION "C")
75
74
(RING)
(RTP DATA CHANNEL OPEN -- TCP CONNECTION "C")
76
(ALERT -- TCP CONNECTION "B")
77
78
(CONNECT-- TCP CONNECTION "B")
(STATUS -- TCP CONNECTION "B")
79
(STATUS -- TCP CONNECTION "A")
80
81
82
RTP IN TCP PACKETS OVER TCP CONNECTION "A"
RTP IN UDP PACKETS OVER RTP DATA CHANNEL
(DTMF --TCP CONNECTION "B")
83
(OUT -- TCP CONNECTION "A")
84
(CLOSE -- TCP CONNECTION "C")
86
85
(RELEASE -- TCP CONNECTION "B")
88
(RELEASE -- TCP CONNECTION "B")
87
(CLOSE -- TCP CONNECTION "C")

FIG. 6

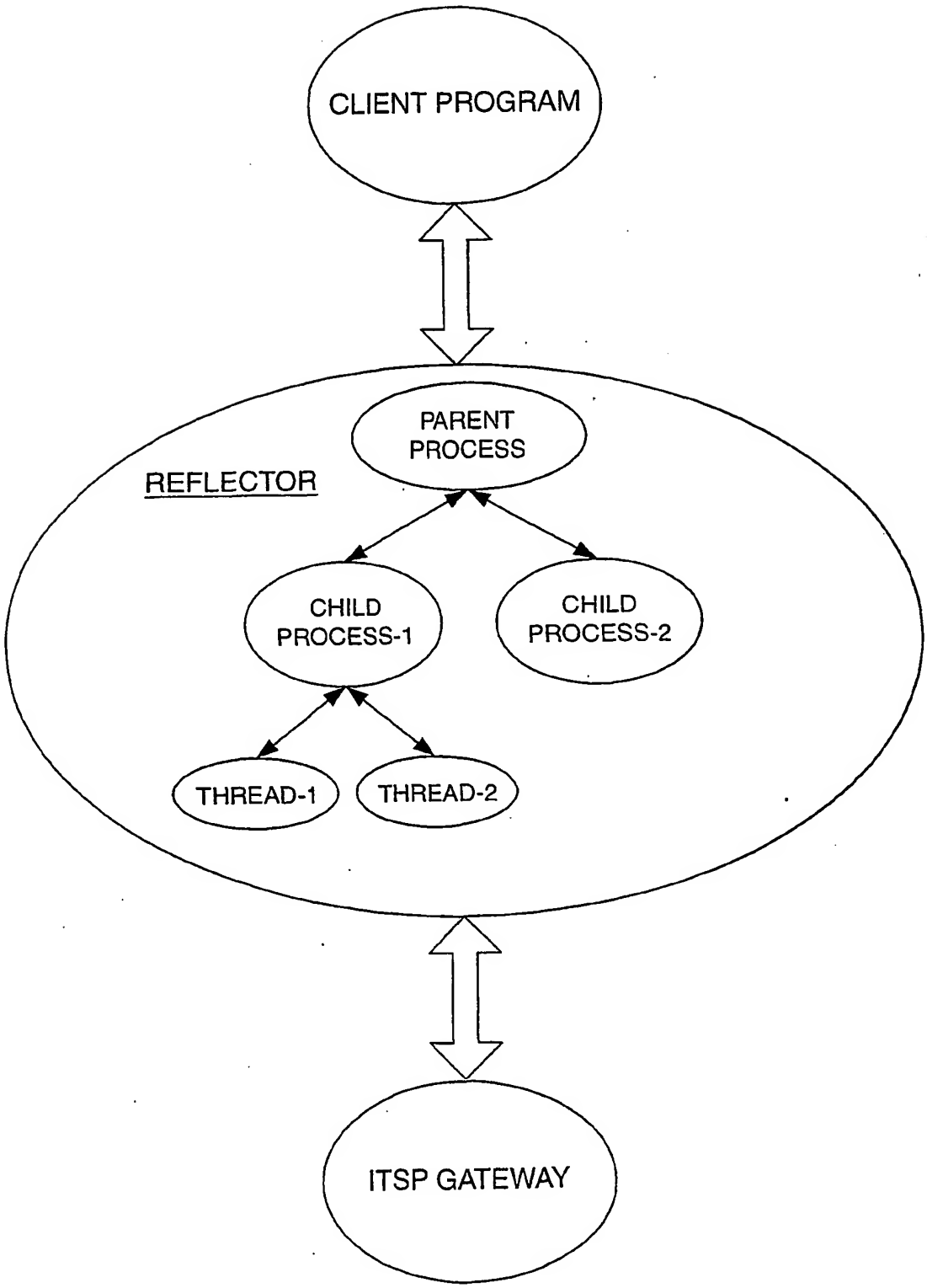

CLIENT PROGRAM
PARENT PROCESS
REFLECTOR
CHILD PROCESS-1
CHILD PROCESS-2
THREAD-1
THREAD-2
ITSP GATEWAY

FIG. 7

INTERNATIONAL SEARCH REPORT

Inter. ional Application No
PCT/US 01/22312

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 858 201 A (SUN MICROSYSTEMS INC) 12 August 1998 (1998-08-12) abstract column 1, line 33 - line 47 column 2, line 19 - line 28 column 4, line 40 -column 5, line 12 column 5, line 40 - line 56 column 10, line 40 - line 54 column 11, line 25 -column 12, line 15 column 13, line 26 - line 57 column 14, line 16 - line 32 ---	1-5,9, 10,12
X	WO 98 42107 A (AT & T CORP) 24 September 1998 (1998-09-24) abstract page 9, line 19 - line 35 claims 2,3,14 --- -/--	1-13

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents:

'A' document defining the general state of the art which is not considered to be of particular relevance

'E' earlier document but published on or after the international filing date

'L' document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

'O' document referring to an oral disclosure, use, exhibition or other means

'P' document published prior to the international filing date but later than the priority date claimed

'T' later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

'X' document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

'Y' document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

'&' document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 February 2002	01/03/2002

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Kesting, V

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/US 01/22312

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 550 984 A (GELB EDWARD J) 27 August 1996 (1996-08-27) abstract	1,2,5,9, 10
A	column 4, line 56 -column 5, line 2 column 5, line 12 -column 6, line 15 column 6, line 41 -column 7, line 3 figure 1	6,7,11
A	BELLOVIN S M ET AL: "NETWORK FIREWALLS" IEEE COMMUNICATIONS MAGAZINE, IEEE SERVICE CENTER. PISCATAWAY, N.J, US, vol. 32, no. 9, 1 September 1994 (1994-09-01), pages 50-57, XP000476555 ISSN: 0163-6804 sections "Packet Filters and UDP", "Tunnels Good and Bad"	1,5-7, 9-11
A	FALKENSTEIN B: "VIDEOKOMMUNIKATION UEBER IP-" NACHRICHTENTECHNIK ELEKTRONIK, VEB VERLAG TECHNIK. BERLIN, DE, vol. 48, no. 5, 1 September 1998 (1998-09-01), pages 20,22-23,25, XP000786648 ISSN: 0323-4657 the entire document	1-12

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inter. ...nal Application No
PCT/US 01/22312

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0858201 A	12-08-1998	EP 0858201 A2	12-08-1998
		JP 2000123097 A	28-04-2000
WO 9842107 A	24-09-1998	US 6075796 A	13-06-2000
		WO 9842107 A1	24-09-1998
		US 6233249 B1	15-05-2001
		US 2001009554 A1	26-07-2001
US 5550984 A	27-08-1996	AU 687575 B2	26-02-1998
		AU 2820295 A	26-06-1996
		CA 2182777 A1	13-06-1996
		CN 1140519 A	15-01-1997
		EP 0744107 A1	27-11-1996
		IL 114178 A	16-08-1998
		JP 3009737 B2	14-02-2000
		JP 9505719 T	03-06-1997
		KR 225574 B1	15-10-1999
		RU 2152691 C1	10-07-2000
		WO 9618253 A1	13-06-1996

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.